Read Performance: The Newest Barrier in Scaled STT-RAM

Yaojun Zhang, Yong Li, Zhenyu Sun, Hai Li, Yiran Chen, and Alex K. Jones

***Abstract*— Spin-torque transfer RAM (STT-RAM), a promising alternative to static RAM (SRAM) for reducing leakage power consumption, has been widely studied to mitigate the impact of its asymmetrically long write latency. However, physical effects of technology scaling down to 45 nm and below, in particular, process variation, introduce the previously unreported and alarming trends in read performance and reliability due to reduced sensing margins and increasing error rates. In this brief, we study the scaling trends of STT-RAM from 65 down to 22 nm as they pertain to read performance, including a 50% increase in sensing versus peripheral circuit delay ratio and a more than 80% increase in uncorrectable read error rates. Through differential sensing, we show how 22 nm can return to sense delay ratio levels at 65 nm and uncorrectable read errors can be reduced by an order of magnitude. Through a case study of a multilevel STT-RAM cache, we show how a reconfigurable cache cell can create an extreme access mode (X-mode) based on differential sensing improve to outperform the state-of-the-art STT-RAM caching techniques in both raw performance and performance per watt by more than 10% while still reducing energy consumption over SRAM caches by more than 1/3.**

***Index Terms*— Memory architecture, nonvolatile memory, STT-RAM.**

I. Introduction

Previous conventional wisdom for spin-torque transfer RAM (STT-RAM) is that writes are slower and require more power than their conventional static RAM (SRAM) counterparts. Several architectural solutions, such as hybrid caches with fast and slow writing memory components [1], various methods for preempting, avoiding, and bypassing writes [2], and leveraging the asymmetry of writing different logic values [3], have been proposed to mitigate the write performance problem. However, due to scaling effects, performance and reliability of STT-RAM reads, not writes, will become the ultimate bottleneck at technologies of 45 nm and below. Read performance, the dominant operation in caches [4], suffers from increased sense amplifier delays for detecting increasingly small sense margins and higher read error rates. In contrast, due to reduced energy barriers at smaller technology nodes, writes will become faster at lower energy, although this leads to higher susceptibility to read disturbance (inadvertent writes from applying a read current).

The primary contribution of this brief is to identify, for the first time, the emerging read performance and stability issue of STT-RAM as the technology node scales. We provide a detailed study on the impact of scaling from 65 to 22 nm, demonstrating a widening performance gap between standard and differential sensing. Furthermore, we present a configurable STT-RAM memory circuit with differential sensing that can selectively operate either in standard mode (S-mode) with a standard slower read performance or in extreme access mode (X-mode) with improved performance and reliability. We demonstrate through a case study that using X-mode in STT-RAM caches can achieve more than 10% improvement in raw performance as well as performance per watt over a design without read optimizations at 22-nm technology.

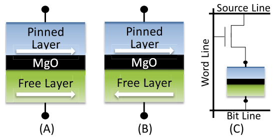


Fig. 1. Illustration of an MTJ and STT-RAM cell.

II. STT-RAM Technology Trends

The building block of STT-RAM is the magnetic tunnel junction (MTJ), which contains two synthetic ferromagnetic layers (pinned and free layer) and one MgO-based tunnel barrier layer [5], as shown in Fig. 1. The magnetic direction of the pinned layer is fixed, while the magnetic direction of the free layer can vary through the application of an external electromagnetic field or spin-polarized current through that layer. When the magnetization directions of the two ferromagnetic layers are parallel, the MTJ is in its low-resistance state [Fig. 1(a)]. In contrast, when the directions of the two layers are antiparallel, the MTJ resistance is high [Fig. 1(b)]. The low and high MTJ resistances can be used to represent logic values. In a typical 1T1J [5] STT-RAM cell shown in Fig. 1(c), one MTJ is connected with one nMOS transistor, which serves as access controller. This nMOS transistor is typically 1.5 times the size of each of the six transistors that comprise an SRAM cell, leading to the four times density improvement assumed in SRAM replacements with STT-RAM [6].

Most existing optimization techniques for STT-RAM have targeted write performance. The relationship between write current (I_c) and write pulsewidth (τ) can be expressed by (1) [7]. Here, I_{c0} is the critical write current at $0K$, τ_{relax} is the relaxation time, and θ_0 is the root-square average of the initial angle of the free-layer magnetization determined by thermal fluctuation

$$I_c(\tau) = I_{c0}\left(1 + \frac{\tau_{\text{relax}}}{\tau} \ln\left(\frac{\pi}{2\theta_0}\right)\right). \quad (1)$$

A. Read Performance Trends

Reads are completed by sensing the voltage differential in the two resistance states of the MTJ using a read current. Even with a relatively small reading current, there is still a probability to flip the MTJ states when applying a read current. Thus, for all reads to MTJs, there is a probability of disturbing the stored value [8].

As the technology scales, the energy required for writing current (I_c) decreases, which is the property that leads to improved writing performance. Unfortunately, this reduction also increases the potential for read disturbance. To avoid increasing the probability of read disturbance, the ratio between read and write current must remain balanced. Fig. 2(a) shows the trend for errors due to read disturbance even as I_c is reduced proportionally with the scaled technology. The trends are alarming, indicating that read disturbance, which is nearly negligible at 65 nm, will reach 0.001% per bit read at 22 nm and will continue to increase as the technology node descends [9].

Manuscript received November 23, 2013; revised February 14, 2014 and May 9, 2014; accepted May 18, 2014. Date of publication July 2, 2014; date of current version May 20, 2015. This work was supported by the National Science Foundation under Award CNS-1116171 and Award CCF-1064976.

The authors are with the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA 15261 USA (e-mail: yaz24@pitt.edu; yol26@pitt.edu; zhs25@pitt.edu; hal66@pitt.edu; yic52@pitt.edu; akjones@pitt.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2014.2326797

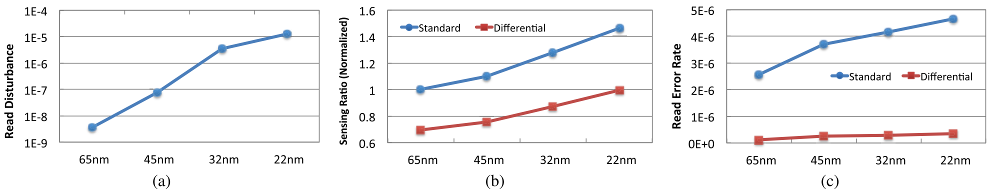


Fig. 2. Read error rate trends with technology. (a) Read disturbance. (b) Ratio of sensing delay to peripheral circuit delay. (c) Uncorrectable read error rate.

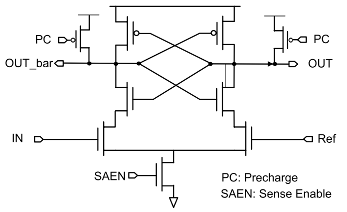


Fig. 3. Sense amplifier design.

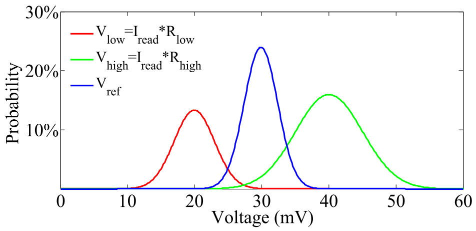


Fig. 4. Read reliability example under variation.

Another method to alleviate read disturbance is to reduce I_r. During a read operation, I_r is applied to the cell, and the resulting voltage difference is compared with a reference voltage (V_{ref}) to differentiate the high- and low-resistance states of a memory cell. The resulting sense margin of the memory cell is proportional to $I_r \cdot \Delta R/2$, where ΔR is the difference between the high- and the low-resistance states. Unfortunately, reducing I_r to mitigate read disturbance also reduces this sensing margin making. The reduced margin makes it more difficult to sense the difference between the two resistance states of the MTJ ultimately increasing sense times and negatively impacting read performance.

Increased process variation at smaller technology nodes exacerbates this problem as performance is heavily impacted by variation, resulting in a distribution of sensing times. Optimistic sense delays reported in [10] often assume the typical case (i.e., the peak of the sense delay curve), implying no significant performance degradation with the reduced sense margins. However, the combination of these two trends—reduced sense margins and increase in variation—create a significant read performance bottleneck.

To demonstrate the impacts of reduced margins and variation, we conducted Monte Carlo simulations using HSPICE of a popular sense-amplifier design in STT-RAM arrays [11] shown in Fig. 3. Only conventional sensing scheme is adopted to offer the highest integration density in STT-RAM design. We note that, nonetheless, the degradation of sense margin will induce the impaction SA designs similar to conventional design. The sensing performance and reliability can be further improved by a better SA design. The sense amplifier was tuned for best possible performance (i.e., the sensing latency) by sizing of the transistors, and timing was derived from an HSPICE simulation. The mean value of high and low resistances are 2 and 1 kΩ, respectively, with a reference resistance of 1.5 kΩ. The assumptions are 5% variation in both transistor size and MTJ shape with an additional 2% intradie variation introduced to account for spatial correlation [12]–[14]. We compared the ratio of sensing time with memory peripheral circuit delays provided through a scaled version of CACTI [15] and report the results in Fig. 2(b) in the series marked standard (normalized to the result for 65 nm). The chart shows that the ratio of the sensing time to the peripheral circuitry delay increases significantly as the technology node descends.

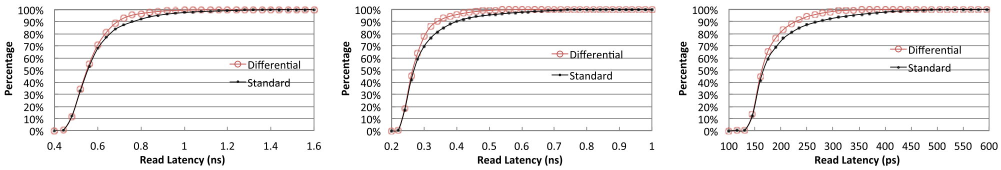
A related trend is uncorrectable read errors due to process variation. The chart in Fig. 4 shows the distributions of comparative voltages for an MTJ in the low-resistance (red) and high-resistance (green) states. Note that this is not actually the input voltage of the sense amplifier. In real simulation, the resistance along the sensing path includes also the contribution from the peripheral circuits, which introduces a steady offset on top of the voltage across the MTJ. These curves are centered around 20 and 40 mV, respectively, but due to process variation, there is a distribution of voltages that would exist. When compared with a reference (blue) centered around 30 mV, to determine the stored value, the red value is expected to be less than the blue, and the green to be higher than the blue. Uncorrectable read errors occur when, due to variation, the cell is in the part of the curves that violates this rule (e.g., far right on the red curve and far left on the blue curve). Our experiments from Fig. 2(c) in the series marked standard show that this error rate is also on an concerning upward trajectory.

While the read disturbance problem has been reported previously in the literature, including a proposed solution to write-back values [9], the relative increase in sensing delay and uncorrectable read error rate due to process variation has not been studied to this point. In the next section, we describe a method to mitigate these undesirable trends.

III. Read Optimization Using X-Mode

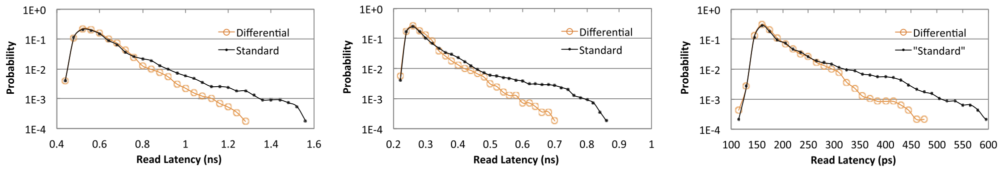
To address the trends of increasing sense time and uncorrectable error rate, we propose to use differential sensing. Differential sensing requires two MTJs to store both the value and its complement, and by sensing the difference rather than comparing with a threshold, one doubles the sense margin ($I_r \cdot \Delta R$). Fig. 2(b) shows the ratio of sensing delay (the sensing delay of the sense amplifier) to peripheral circuit delay (including routing, decoding, and I/O) of differential sensing, which is dramatically reduced with respect to the standard sensing. At 22 nm, the ratio is similar to the standard sensing ratio at 65 nm. In addition, the increase in sense ratio is less steep, providing some relief from this negative performance trend.

Furthermore, many uncorrectable errors for an individual MTJ can be avoided using differential access, as shown in Fig. 2(c). Here, the read error rate is defined as the probability that the output of the sense amplifier is different from the one stored in the memory cell. At 65 nm, the error rate is reduced by an order of magnitude using differential sensing. The error rate increase trend for differential access is nominal as technology scales, whereas the standard MTJ error-rate increase is more dramatic. This impact can be explained by again considering Fig. 4. For a standard access, the errors occur at the intersection of the red and blue curves (stuck at 1) and the green and blue curves (stuck at 0). For a differential access, the errors occur at the intersection of the red and green curves of the two MTJs. As process variation has spatial locality, in the cases where physically adjacent cells are used to store the differential values, this probability of overlap is further reduced.



(a) (b) (c)

Fig. 5. Cumulative distribution functions of sense amplifier latency for standard and differential reads. (a) 45 nm. (b) 32 nm. (c) 22 nm.



(a) (b) (c)

Fig. 6. Sense amplifier latency distribution to ensure $<10^{-4}$ read error rate due to violating the required delay. (a) 45 nm. (b) 32 nm. (c) 22 nm.

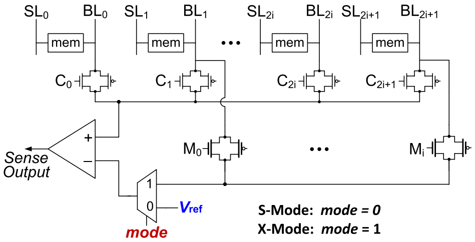


Fig. 7. Reconfigurable extreme mode (X-mode) memory circuit.

Fig. 5 shows the detailed cumulative distribution functions for the sense amplifier performance from 45- down to 22-nm technology. For all technologies, the differential access curve shows a sharper trend to achieve coverage at a smaller latency. Thus, the minimum latency to achieve full coverage across all STT-RAM cells in the die is lower for differential sensing than standard sensing. Fig. 6 shows this latency difference as the technology node scales when tolerating an error rate of 10^{-4}. The trends indicate that as the technology node descends, the performance improvement using differential sensing increases, reaching a 25% speedup at 22 nm.

Differential sensing has two main drawbacks. The area required to store a value is doubled, and the dynamic power of write access is also effectively doubled. To efficiently leverage differential sensing and to minimize these side effects, we propose a configurable differential cell shown in Fig. 7. Through a MUX and transmission gate, the cell can be configured into standard high-density mode (S-mode) by comparing the selected cell with V_{ref} or in situations where read performance is critical to use extreme performance mode (X-mode) by sensing the voltage difference between adjacent cells. Note that the transmission gate (M_i) is shared by multiple bitlines. The direct overhead of each bitline is only four transistors, which is negligible in the STT-RAM design incurring very minimum area overhead in our design. Although such a scheme will increasing the probability of read disturbance, since the read disturbance error is exponential related to the reading current as we mentioned, a small reduction of reading current could significantly reduce the disturbance error. Since this scheme can largely reduce the sensing error, with a smaller reading current, it can also reduced the read disturbance error without sacrificing the sensing error.

In Table I, we report the read access latencies of memories of different sizes that employ the configurable cell design from Fig. 7 for four memory sizes consistent with cache banks at different cache levels (e.g., a 32K L1 or a 4M L3 bank). In standard S-mode, the sense delay reports 99.9% of the tail of the delay distribution similar to the curves from Fig. 6. The sensing time steadily increases from 58% at 65 nm to 67% at 22 nm of the total access time of an L1-sized memory. Similarly, the sensing time is significant and increases from 22% to 32% for a fairly large L3 tile sized memory. Extreme X-mode access provides a significant speed benefit providing more than 20% access latency improvement for an L1 cache at 22 nm. Thus, X-mode can be used to maintain a competitive read performance with SRAM. However, in cases where dynamic power and capacity are critical, S-mode still can be used effectively.

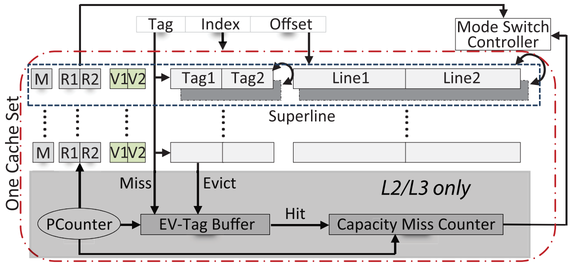


Fig. 8. Configurable cache architecture (X-cache).

IV. Case Study: X-Cache Design

Based on the technology trends and the configurable memory cell from Section III, we present a case study of a configurable cache architecture. X-cache that can be adapted dynamically at the cache block granularity based on application needs for faster read performance or higher capacity. Two adjacent partner lines (Line1 and Line2) within the X-cache set (Fig. 8) form one superline. Each superline contains two respective tags (Tag1 and Tag2) and valid bits (V1 and V2) and one mode bit (M), indicating whether the partners are storing one value in X-mode or two independent values in S-mode.

To perform efficient mode switching, each line is instrumented with a reference counter (R1/R2), which tracks the amount of accesses per line. Using a static threshold, heavily read lines can be promoted to X-mode, while heavily written or less frequently accessed lines can remain in S-mode. For lower levels of the cache (e.g., L2/L3), which are typically more sensitive to capacity, additional steps are taken to ensure that lines promoted to X-mode do not significantly increase miss rate, and by extension, harm performance.

A. L1 Design

To build an effective STT-RAM L1 cache, nearly all read accesses in L1 must occur in X-mode to be competitive with the performance of SRAM. X-mode reads do not require additional dynamic power as the read current is still only injected once for a differential comparison. However, unlike higher levels in the cache, the number

TABLE I
IMPACT OF SENSING DELAY ON READ PERFORMANCE WITH TECHNOLOGY SCALING FOR DIFFERENT CACHE SIZES

	Cache Configuration	32K 4-Way		64K 4-Way		256K 8-Way		4M 16-Way	
Tech.	Memory Type/Mode	STT S	STT X	STT S	STT X	STT S	STT X	STT S	STT X
22nm	SenseAmp (ns)	0.47	0.32	0.47	0.32	0.47	0.32	0.47	0.32
	Total (ns)	0.699	0.549	0.749	0.599	0.963	0.813	1.485	1.335
	Sensing Ratio	67.2%	58.3%	62.8%	53.4%	48.8%	39.4%	31.6%	24.0%
32nm	SenseAmp (ns)	0.76	0.52	0.76	0.52	0.76	0.52	0.76	0.52
	Total (ns)	1.184	0.944	1.316	1.076	1.731	1.491	2.766	2.526
	Sensing Ratio	64.2%	55.1%	57.8%	48.3%	43.9%	34.9%	27.5%	20.6%
45nm	SenseAmp (ns)	1.21	0.83	1.21	0.83	1.21	0.83	1.21	0.83
	Total (ns)	0.786	0.786	0.99	0.99	2.861	2.481	4.904	4.524
	Sensing Ratio	60.6%	51.4%	55.0%	45.6%	42.3%	33.5%	24.7%	18.3%
65nm	SenseAmp (ns)	1.51	1.05	1.51	1.05	1.51	1.05	1.51	1.05
	Total (ns)	2.587	2.127	2.922	2.462	3.803	3.343	6.787	6.327
	Sensing	58.4%	49.4%	51.7%	42.6%	39.7%	31.4%	22.2%	16.6%

TABLE II
ARCHITECTURAL PARAMETERS

Basics	16 cores, 2 issue width, 3.5GHz CPUs			64-bit Solaris 10 OS			4GB main memory, 150-cycle latency		
Caches	Private L1 Cache (MESI)			Private L2 Cache			Shared L3 Cache		
	32K 4-way 64B blk			256K 8-way 64B blk			16M 16-way 64B blk	64M 16-way 64B blk	
	SRAM	STT S	STT X	SRAM	STT S	STT X	SRAM	STT S	STT X
Size (mm^2/core)	0.048	0.031		0.233	0.085		0.96	1.006	
Read Latency (cycles)	2	3	2	4	4	3	5	6	5
Write Latency (cycles)	3	5	5	3	26	26	4	27	27
Read Energy (nJ)	0.029	0.014	0.014	0.032	0.022	0.022	0.054	0.046	0.046
Write Energy (nJ)	0.031	0.094	0.188	0.036	0.117	0.234	0.06	0.26	0.52
Leakage Power (mW)	149	63	63	664	138	138	1249	471	471
X-cache parameters	RCounter: 2 bits, M/PCounter: N/A			RCounter: 12 bits, MCounter: 8 bits, PCounter:20 bits, EV-Tag Buffer: 1-entry					
Thresholds	X-mode promotion: 4 consecutive reads			MTSX: 6, RTSX: 12, MTXS: 2, RTXS: 9					

of L1 writes is significant—nearly 20% of accesses on average. While write performance in X-mode is not degraded in comparison with S-mode as both the cell and its complement can be written in parallel, writes to STT-RAM even with reduced retention times still require a significantly higher dynamic power than equivalent technology SRAM and about twice as much dynamic power than S-mode writes.

Our analysis of application phases indicates that memory locations with heavy write activity during a phase are infrequently read. During a later phase, the same memory location can be frequently read with minimal writing. Based on this analysis, the X-cache L1 superline always stores only one line. All writes are conducted initially in S-mode leaving the partner line invalid. After a number of consecutive reads to the same location, the line is promoted to X-mode for further reads as follows. The reference counter (R1) accumulates during each read, and upon reaching a set threshold after storing the complement in the adjacent cell, sets the mode bit (M) to X-mode. Upon a write, the counter is reset and M sets to S-mode. This allows intermittent reading and writing to predominantly occur in S-mode, but many successive reads will be serviced in X-mode. The L1 cache requires only one tag, reference counter, and valid bit per superline as when the superline is in S-mode one line is left empty.

B. L2/L3 Design

The primary goal of the higher-level X-cache design is to accelerate as many read accesses as possible while retaining the largest possible overall capacity. Thus, heavily accessed locations are set to X-mode *ad hoc*, while the majority remain in S-mode to preserve capacity. In contrast to L1, writes are permitted in X-mode as writes to last-level cache (LLC) comprises <1% of accesses.

As shown in Fig. 8, to promote an S-mode block to X-mode block, the LRU block needs to be evicted and the adjacent block of the S-mode block needs to be relocated to the LRU block. Then, the complementary values of the S-mode block can be programmed into the adjacent block to form an X-mode block. We use reference and miss counters to efficiently reconfigure the working modes of cache blocks and minimize the overheads. The most recently evicted tag is retained in the EV-Tag buffer, which maintains a capacity miss counter (CMC) for each cache set. On a read miss that hits in the EV-Tag buffer, the CMC is incremented. This allows the system to evaluate the performance improvement of the X-mode promotion compared with the additional misses resulting from the reduced capacity. The CMC and the eviction tag buffer collect application specific information at runtime to make cell configuration decisions. For example, differential sensing cells can be employed for a cache block only when the cache pressure is low in the corresponding cache set and the read frequency for that block is high. From our experiments, heavily read data elements are concentrated within a very small percentage of all cache blocks, making this tradeoff extremely effective.

V. EVALUATION

We use Wind River Simics [16] to simulate a 16-core CMP with the cache architecture, as described in Table II, based on the 22-nm latencies presented in Table I and 22-nm SRAM parameters from a scaled version of CACTI [17].

We compare the X-cache scheme with an SRAM-only design and the leading STT-RAM technique of all S-mode cells with reduced retention time (26.5 μs) and dynamic ECC data correction in the L1 cache (STT-RR) [18]. All STT-RAM designs use a 64M (4M/core) LLC, while the SRAM cache uses a 16M (1M/core) LLC for a same die area comparison, similar to the architecture evaluated in [18]. The scheme XC-L1 applies the X-cache technique on top of STT-RR to optimize read performance for L1 only with L2/L3 in all S-mode configuration. XC-A further extends X-mode for heavily read locations at the L2 and L3 (all) cache levels. Our input workloads consist of parallel C/C++ and Java benchmarks from the DACAPO [19], SPLASH-2 [20], and PARSEC [21] benchmark suites.

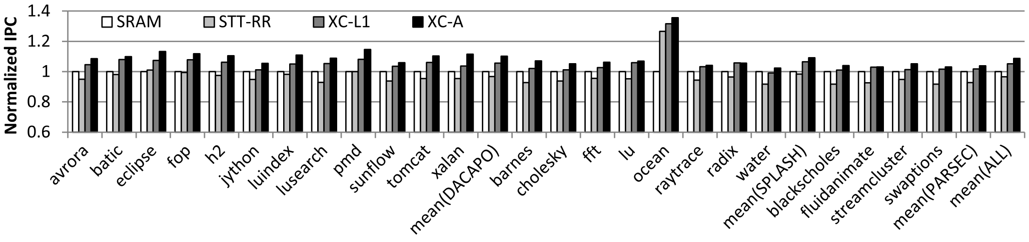


Fig. 9. Performance (IPC) comparison of SRAM and STT-RAM with X-cache (normalized to SRAM).

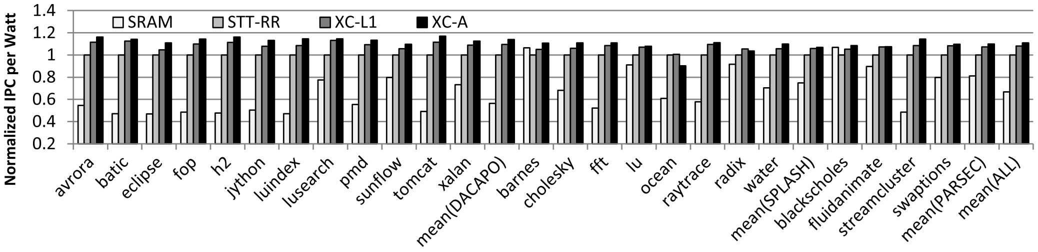


Fig. 10. Performance per watt of X-cache (normalized to STT-RR).

The performance comparison, presented as instructions per cycle (IPC) normalized to SRAM, is shown in Fig. 9. STT-RR performs poorly compared with SRAM in spite of the capacity advantage due to the increased read latency at L1. In contrast, XC-L1 alone provides a 5% improvement over SRAM and a 9% improvement over STT-RR on average. Applying X-cache to all levels, XC-A outperforms SRAM by 9% and STT-RR by 13% on average.

The overall benefit of X-cache is shown in Fig. 10, which presents the IPC/watt of X-cache normalized to STT-RR. On average, XC-L1 and XC-A provide a 8% and 11% improvement, respectively, over STT-RR in IPC/watt, and XC-L1 and XC-A provide a 75% and 80% improvement on average, respectively, over SRAM in IPC/watt.

VI. Conclusion

In this brief, we identify several read performance bottlenecks in scaled STT-RAM cells. We provide a detailed study of the proportional increases in sensing delay and read error rates, and demonstrate that differential sensing is a viable solution to minimize these impacts as technology nodes scale. Using a proposed runtime configurable extreme access mode (X-mode) memory cell, we demonstrate in a case study how differential access can provide better than 10% performance and performance per watt improvement over the state-of-the-art STT-RAM cache designs in 22-nm technology while reducing total power consumption compared with the traditional SRAM caches by 34%.

References

[1] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," in *Proc. 36th Annu. ISCA*, 2009, pp. 34–45.

[2] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for STT-RAM using early write termination," in *Proc. IEEE/ACM ICCAD*, Nov. 2009, pp. 264–268.

[3] M. Qureshi, M. Franceschini, A. Jagmohan, and L. Lastras, "PreSET: Improving performance of phase change memories by exploiting asymmetry in write times," in *Proc. 39th ISCA*, Jun. 2012, pp. 380–391.

[4] B. Amrutur and M. Horowitz, "Speed and power scaling of SRAM's," *IEEE J. Solid-State Circuits*, vol. 35, no. 2, pp. 175–185, Feb. 2000.

[5] M. Hosomi *et al.*, "A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-ram," in *IEEE IEDM Tech. Dig.*, Dec. 2005, vol. 2, no. 25, pp. 459–462.

[6] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A novel architecture of the 3D stacked MRAM L2 cache for CMPs," in *Proc. IEEE 15th Int. Symp. HPCA*, Feb. 2009, pp. 239–249.

[7] S. Kang and K. Lee, "Emerging materials and devices in spintronic integrated circuits for energy-smart mobile computing and connectivity," *Acta Mater.*, vol. 61, no. 3, pp. 952–973, 2013.

[8] J. Li, P. Ndai, A. Goel, S. Salahuddin, and K. Roy, "Design paradigm for robust spin-torque transfer magnetic RAM (STT MRAM) from circuit/architecture perspective," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 12, pp. 1710–1723, Dec. 2010.

[9] Z. Sun, H. Li, and W. Wu, "A dual-mode architecture for fast-switching STT-RAM," in *Proc. ACM/IEEE ISLPED*, Jul. 2012, pp. 45–50.

[10] C. Smullen, IV, V. Mohan, A. Nigam, S. Gurumurthi, and M. Stan, "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," in *Proc. 17th Int. Symp. HPCA*, Feb. 2011, pp. 50–61.

[11] C.-T. Cheng, Y.-C. Tsai, and K.-H. Cheng, "A high-speed current mode sense amplifier for spin-torque transfer magnetic random access memory," in *Proc. 53rd IEEE Int. MWSCAS*, Aug. 2010, pp. 181–184.

[12] J. Li, C. Augustine, S. Salahuddin, and K. Roy, "Modeling of failure probability and statistical design of spin-torque transfer magnetic random access memory (STT MRAM) array for yield enhancement," in *Proc. 45th ACM/IEEE DAC*, Jun. 2008, pp. 278–283.

[13] Y. Chen, H. Li, X. Wang, W. Zhu, W. Xu, and T. Zhang, "Combined magnetic- and circuit-level enhancements for the nondestructive self-reference scheme of STT-RAM," in *Proc. ACM/IEEE ISLPED*, Aug. 2010, pp. 1–6.

[14] J. Z. Sun. (2000). Spin-current interaction with a monodomain magnetic body: A model study. *Phys. Rev. B* [Online] *62(1)*, pp. 570–578. Available: http://link.aps.org/doi/10.1103/PhysRevB.62.570

[15] P. Shivakumar and N. P. Jouppi, "CACTI 3.0: An integrated cache timing, power, and area model," hp, Palo Alto, CA, USA, Tech. Rep. WRL-2001-2, Aug. 2001.

[16] P. S. Magnusson *et al.*, "Simics: A full system simulation platform," *IEEE Comput.*, vol. 35, no. 2, pp. 50–58, Feb. 2002.

[17] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi. (2008). CACTI 5.1, Tech. Rep. HPL-2008-20 [Online]. Available: http://www.hpl.hp.com/research/cacti/

[18] Z. Sun *et al.*, "Multi retention level STT-RAM cache designs with a dynamic refresh scheme," in *Proc. 44th Annu. IEEE/ACM Int. Symp. Microarchit.*, Dec. 2011, pp. 329–338.

[19] S. M. Blackburn *et al.*, "The DaCapo benchmarks: Java benchmarking development and analysis," *ACM SIGPLAN Notices*, vol. 41, no. 10, pp. 169–190, 2006.

[20] J. M. Arnold, D. A. Buell, and E. G. Davis, "Splash 2," in *Proc. 4th Annu. SPAA*, 1992, pp. 316–332.

[21] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," Dept. Comput. Sci., Princeton Univ., Princeton, NJ, USA, Tech. Rep. TR-811-08, Jan. 2008.